UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/747,659	12/26/2003	Pat Styles	13768.783.118	1572

47973 7590 12/31/2008
WORKMAN NYDEGGER/MICROSOFT
1000 EAGLE GATE TOWER
60 EAST SOUTH TEMPLE
SALT LAKE CITY, UT 84111

EXAMINER	
LEE, MARINA	
ART UNIT	**PAPER NUMBER**
2192	
MAIL DATE	**DELIVERY MODE**
12/31/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No. 10/747,659	Applicant(s) STYLES ET AL.	
	Examiner MARINA LEE	Art Unit 2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *30 October 2008*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-4,6-16 and 18* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-4, 6-16, and 18* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on October 30, 2008 has been entered.

2. This action is responsive to Amendment filed on October 30, 2008. Claims 1 and 11 have been amended. Claims 5 and 17 have been cancelled. Claim 18 has been newly added. Thus, Claims 1-4, 6-16, and 18 are presented for examination.

Prior Art's Arguments – Rejections

3. Applicant's arguments filed on October 30, 2008, especially on pages 10-13 of Remarks, with respect to claim limitation, "while compiling the source code file, extracting information that identifies a location of the source code file, the version associated the source code file, a name of the server, a port of the server at which the server may be accessed to access the source code, a path to the source code, and a numeric value that indicates a version number of the sourced", recites per claims 1 and 11 respectively, have been fully considered and are persuasive, however, after further consideration, the new ground of rejection is being applied in view Baisley et al., (U.S. Patent No. 6,106,574 of record) in view of view of Haikin (U. S. Patent No. 6,757,893 B1 of record), Murakami et al. (US 2003/0167423 A1 made of record) and Boxall et al. (US

6263,456, B1 made of record) as will be further addressed under the item (5) below.

Claim Objections

4. Claim 16 is objected to because of the following informalities:

As to claim 16, recites the repeated limitation of claim 11. Thus, claims 16 should be deleted. Appropriate correction is required.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claims 1-4, 6-16, and 18 are rejected under 35U.S.C. 103(a) as being unpatentable over Baisley et al., (U.S. Patent No. 6,106,574 made of record– hereinafter Baisley), in view of Haikin (U. S. Patent No. 6,757,893 B1 of record), Murakami et al. (US 2003/0167423 A1 made of record – hereinafter Murakami), and Boxall et al. (US 6263,456, B1 made of record – hereinafter Boxall).

As per claims 1 and 11, Baisley discloses a method for associating original source code with binary code for debugging the binary code, the method comprising:

storing a source code file (*source files 22, Fig. 1, which contains program elements (e.g., function 23...), -- See at least col. 5: 14-17 and col. 1: 24-28*), the source code file including source code and being associated with a version (*e.g., source file name – see at least col. 5: 28-30 with emphasis added*);

compiling the source code file into a binary file (e.*g., binary file 36, Fig. 1) – (e.g., compiler 20 convert the source files 22 into objects 30-34, Fig. 1, which represent machine/binary instruction – see at least col. 5: 17-22 with emphasis added*) ;

while compiling the source code file, extracting information (*e.g. table mapping instructions to source code location*) that identified a location of the source code file and the version (*e.g., source file name*) associated with the source code file – *see at least col. 5: 20-30* ;

storing the extracted information in a debug file associated with the binary file – *see at least col. 5: 39-42 with emphasis added*;

after compiling the source code file, receiving an instruction for a debugger to debug the binary file – *see at least col. 5: 39-51*;

after receiving the instruction for the debugger, using the extracted information in the debug file, locating the source code file and associate it with the binary file – (*e.g., a debugger uses table to find the source location such as source file name and a line number – see at least col. 5: 45-51 & 28-30 with emphasis added*); and thereafter debugging the binary file with full source code support by correlating lines of the source code file with binary instructions in the binary file, the source code file including only the source code originally used to compile the binary file (*e.g., a debugger uses table to fined the source location such as source file name and a line number – see at least col. 5: 45-51 & 28-30 with emphasis added)*.

It is noted that Baisley does not explicitly disclose storing a source code file (*source files 22, Fig. 1, which contains program elements (e.g., function 23...), -- See at least col. 5: 14-17 and col. 1: 24-28*) on a server, the source code file including source code and being associated with a version. The extracting information identifying the version associated with the source code file, a numeric number that indicates a version number of the source code.

However, Haikin, in an analogous art, teaches a software source code version control system for use during the development and maintenance of a software system by multiple software developer in which historical version tracking is maintained for all source code on a line-by-line basis <u>on central server 20, Fig. 1</u>, without requiring excessive storage area, in which source code can be accessed and modified by more than one software developer at a time, in which historical version tracking of a broad functional changes is provided and in which <u>quick and transparent access</u> is provided to <u>each version of the source code</u> (e.g., 2.0) (*see Haikin, at least col. 3: 21-31, col. 7: 20-42, and col. 8: 23-59 with emphasis added*).

It would have been obvious to one ordinary skill in the art at the time the invention was made to use the source code version control system of Haikin in source code files 22 (fig. 1) of Baisley for providing storing of each version the source code and providing quick access and maintaining of the historical version for all source code without requiring excess storage space as taught in Haikin (*e.g., col. 3: 1-18*).

As indicate above, Baisley while compiling the source code file, extracting information (*e.g. table mapping instructions to source code location*) that identified a location of the source code file and the version (*e.g., source file name*) associated with the source code file – *see* Baisley*, at least col. 5: 20-30.*

It is to note that modified Baisley and Haikin do not explicitly disclose extracting information that included a name of server, a path to the source code. However, Murakami, in an analogous art, teaches testing consistency of machine code file and source files. "[t]he computer system compiles a source code in the source file 1 into a machine code 3a (step S1), thus producing a machine code file 3. At the same time, the computer system collects file attributes (step s2) of the source file 1 (e.g., source file attribute 32 (fig. 10) such as i.e. location: http//sv1/mas/sorc/pag.c …), which includes, for example, the following information: file location 2a, last modified date 2b, and file size 2c. The file location 2a indicates where the source file 1 is stored. The last modified date 2b shows when the source file 1 was modified last time" – *See Murakami, at least, [0035], [0079], and (source file attribute 32) of Fig. 10, with emphasis added.*

It would have been obvious to one ordinary skill in the art at the time invention was made to use the source file attribute (e.g. file 32 of Fig. 10) of Murakami in the mapping table of the modified Baisley and Haikin for optimizing consistency between the debugging files with the source files as taught in Murakami *(e.g., [0002]).*

It is to note that modified Baisley with Haikin, and Murakami do not explicitly disclose extracting information that includes a port number of the server at which the server may be accessed to access the source code. However, Boxall, in an analogous art, teaches remote debugging system for client/server application. By providing parameters passing to the remote debugger UI 19, which the parameter includes … <u>a port number</u> identifying the port on which the debugger UI is listening/accessing. – *See Boxall, at least, title, and col. 6: 43-52 with emphasis added.*

It would have been obvious to one ordinary skill in the art at the time invention was made to include parameter (e.g., port number) of Murakami in the mapping table of the modified Baisley with Haikin for quickly remote access location to each individual version source code during mapping between the debugging files with the source files.

It is further to note that modified Baisley with Haikin, Murakami, and Boxall, does not explicitly disclose *a computer-readable storage medium* having computer executable instructions stored thereon that, when executed by a processor, implemented above method. However, Haikin discloses a computer storage medium (e.g., removable CD-ROM media – see at least col. 7: 36).

It is well known in the computer art that such method step can be implemented as computer program and be known, commonly practiced and/or stored on the removable CD-ROM media of Haikin for implemented the above method. Thus, it would have been obvious in view of reference teachings above.

Further regarding to claim 11, Haikin also discloses a system (*e.g., workstation 30 (Fig. 3) – see at least col. 8: 46-50*) for implementing the above method.

As to claim 12, modified Baisley with Haikin discloses further comprising a source server (e.g., *source code storage 270 (fig.2)— see Haikin at least col. 8: 24-45)* arranged to extract the information (*e.g., table mapping of Baisley*) at debug time, retrieve the source code files from the version control server, and place the source code files in a directory accessible by the debugger – (*e.g., a debugger uses table to find the source location such as source file name and a line number – see Baisley, at least col. 5: 45-51 & 28-30 with emphasis added*).

As to claim 2, modified Baisley with Haikin discloses further comprising:

extracting the information from the debug file –*see Baisley, at least col. 5: 39-42 with emphasis added*;

requesting the source code associated with the version from the server via the information (*see Baisley, at least col. 5: 20-30)*;

placing the source code in a directory used by a debugger to debug the executable code –*see Baisley, at least col. 5: 39-42 with emphasis added*; and executing the debugger and matching an instruction in the executable code to an instruction in the source code (*e.g., a debugger uses table to find the source location such as source file name and a line number – see Baisley, at least col. 5: 45-51 & 28-30 with emphasis added*).

As to claim 3, modified Baisley with Haikin also discloses wherein the source code file includes programming statements (*source files 22, Fig. 1, which contains program elements (e.g., function 23...), -- See Baisley, at least col. 5: 14-17 and col. 1: 24-28)*, which, when compiled, produce executable code in the form of the binary file – *(e.g., compiler 20 convert the source files 22 into objects 30-34, Fig. 1, which represent machine/binary instruction(*e.g., *binary file 36, Fig. 1) – see Baisley, at least col. 5: 17-22 with emphasis added*)

As to claim 4, modified Baisley with Haikin further discloses wherein the server comprises a version control server (e.g., version control server module 280, Fig. 2) that stores a plurality of versions of the source code – *See Haikin, at least col. 8: 38-42.*

As to claim 6, modified Baisley with Haikin also discloses wherein the binary file includes code that was compiled form a plurality of source code files (e.g., FILE1, FILE 2...), each source code file associated with a version (e.g., file name such as FILE 1, FILE 2..) – *See Baisley, at least col. 8: 38-67 and col. 9: 1-42 with emphasis added.*

As to claim 7, modified Baisley with Haikin also discloses further comprising obtaining additional information that identified the version associated with the plurality of source code files to the server and storing the additional information in the debug file – *See Haikin, at least col. 7: 20-42, col. 8: 38-42, col. 10: 26-30, and col. 16: 12-50 with emphasis added*.

As per claims 8 and 14, modified Baisley with Haikin further discloses wherein the debug file comprises a program database file (e.g*., source code storage 270 (fig.2) — see Haikin at least col. 8: 24-45*) that is separate form the executable code (*e.g., binary file 36 (fig. 1) contains machine code instruction – see Baisley at least col. 5: 17-22*).

As to claim 9, modified Baisley with Haikin also discloses wherein the debug file comprises a portion of an executable file (e.g., machine instruction) that includes the executable code – *see Baisley, at least col. 5: 17-22 & 42-44*.

As to claim 10, modified Baisley with Haikin disclose further comprising

iterating each source code file that is part of a compilation, each source code file having a version (*see Baisley, at least col. 5: 14-22*);

obtaining information that identified the version of each source code file to the server and a local name of each source code file – *See Haikin, at least col. 7: 20-42, col. 8: 38-42, col. 10: 26-30, and col. 16: 12-50 with emphasis added*.;

storing the information in a lookup table (*e.g. table mapping instructions to source code location -See Baisley, at least col. 5: 20-30*); and

extracting, from the binary file, local names of the source code files that were used in compiling the binary file – *e.g., a debugger uses table to find the* *source location such as source file name and a line number – see Baisley, at least col. 5: 45-51 & 28-30 with emphasis added*; and

for each source code file that was used in compiling the binary file, looking up the version in the lookup table using the local name of the source code file – *See Baisley, at least col. 8: 38-67 and col. 9: 1-42 with emphasis added*.

As to claim 13, modified Baisley with Haikin discloses wherein the source server comprises a component of the debugger (e.g., *table mapping – see Baisley, at least col. 5: 45-51 & 28-30 with emphasis added).*

As to claim 15, modified Baisley with Haikin discloses wherein the debugger is arranged to find the source code files in the directory and is unaware of the version control server (e.g., *e.g., a debugger uses table to find the source location such as source file name and a line number – see Baisley, at least col. 5: 45-51 & 28-30 with emphasis added).*

Claim 16 recites repeated limitation of claim 11 above. Accordingly claim16 is being applied the same rejection as of claim 11 indicated above.

As to claim 18, Modified Baisley with Haikin, Murakami, and Boxall discloses wherein placing the source code in a directory used by the debugger to debug the executable code is performed after launching the debugger, and wherein the method further discloses:

storing in the debug file information from a data stream (*i.e. location: http: //sv1/mas/sorc/pag.c …*) and when the data stream is divided into a global variable area (e.g. http,) , a local variable area (master/server) , and a source files area (e.g., pg.c) , the data stream including variable in a shorthand expressions requiring an extractor substitute the variables with an expression or value identifying information necessary for retrieval of the sour code file (e.g., inserting the source file attribute 32 into XML file) ; and

merging the debugging file with the binary file *–see Baisley, at least col. 5: 39-42 with emphasis added*.

Conclusion

7. The prior art made of record and not relied upon, which is cited on (form 892) is considered pertinent to application disclosure.

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Marina Lee whose telephone number is (571) 270-1648. The examiner can normally be reached on M-F (11:00 am to 7: 30 pm) EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/M. L./	/Tuan Q. Dam/
Examiner, Art Unit 2192	Supervisory Patent Examiner, Art Unit 2192